

In the Claims:

Please cancel Claims 5, 12, 19; and amend Claims 1-4, 6-11, 13-18 and 20-21, all as shown below. Applicant respectfully reserves the right to prosecute any originally presented claims in a continuing or future application.

1. (Currently Amended) A system that provides a generic user interface testing framework, and allows a user to test and debug graphical user interfaces for software applications under development, comprising:

a computer including a computer readable medium, and a processor operating thereon;

a software application source code, stored on the computer readable medium, wherein the software application source code defines a software application under development, including a graphical user interface as part of the software application, and wherein the software application source code executes on the computer to display its graphical user interface;

one or more software test tools that are invoked to perform testing operations on the graphical user interface that is displayed while the software application is running;

a test case input file stored on the computer readable medium, that contains a plurality of generic interface commands, wherein the test case input file can be edited and reused as necessary by a user to specify different generic interface commands for testing against a software application's graphical user interface in the same or a different software test tool; and

an interpretive engine that executes on the computer, and that includes a plurality of dynamically loaded libraries corresponding to the plurality of software test tools, wherein the interpretive engine receives the generic interface commands defined in the test case input file, loads required libraries to map the generic interface commands to corresponding tool-specific testing operations, invokes the software test tools to perform the testing operations on the software application's graphical user interface, and reports the success or failure of the testing operations that receives and translates said generic interface commands from a tester; and

a native library to native language understood by a particular test software tool;

wherein the interpretive engine uses the native library

wherein libraries can be loaded to support different test software suites and applications.

2. (Currently Amended) The system of claim 1 wherein the system includes the ~~[[test]]~~ software test tools stored locally on a computer processing system containing the user interface testing framework.
3. (Currently Amended) The system of claim 1 wherein ~~[[test]]~~ software test tools ~~[[is]]~~ are stored at another computer processing system or machine.
4. (Currently Amended) The system of claim 1 wherein ~~[[the]]~~ an editor or wizard provides a graphical interface to allow the user tester to ~~enter said test~~ edit or create the test case input file.
5. (Canceled).
6. (Currently Amended) The system of claim 1 wherein the test case input file ~~commands~~ can be is created offline and subsequently communicated to the interpretive engine.
7. (Currently Amended) The system of claim 1 wherein any of the ~~test~~ software test tools can be removed and replaced with another test software test tool.
8. (Currently Amended) A method for providing a generic user interface testing framework that allows a user to test and debug graphical user interfaces for software applications under development, comprising the steps of:
 - executing a software application source code, wherein the software application source code defines a software application under development, including a graphical user interface as part of the software application, and wherein the software the software application source code executes to display its graphical user interface;
 - providing one or more software test tools that are invoked to perform testing operations on the graphical user interface that is displayed while the software application is running;
 - allowing a user tester to enter a test case input file stored on the computer readable medium, that contains a plurality of generic interface commands, wherein the test case input file can be edited and reused as necessary by a user to specify different generic interface

commands for testing against a software application's graphical user interface in the same or a different software test tool; and number of commands via an editor or interface; and

translating, using an interpretive engine using a plurality of dynamically loaded libraries corresponding to the plurality of software test tools to receive the generic interface commands defined in the test case input file, load required libraries to map the generic interface commands to corresponding tool-specific testing operations, invoke the software test tools to perform the testing operations on the software application's graphical user interface, and report the success or failure of the testing operations, the generic interface commands received from the tester, and mapping, using a native library, the generic commands to native language understood by a particular test software tool;

wherein the interpretive engine uses the native library to map the into tool-dependent codes that are then passed to the test software tool,

wherein libraries can be loaded to support different test software suites and applications.

9. (Currently Amended) The method of claim 8 wherein the ~~system includes the test software test tools~~ are stored locally on [[the]] a same computer or machine as the software application under development.

10. (Currently Amended) The method of claim 8 wherein the ~~test software test tools~~ [[is]] are stored at another computer or machine as the software application under development.

11. (Currently Amended) The method of claim 8 wherein the ~~an editor or wizard~~ provides a graphical interface to allow the user tester to enter said test commands to edit or create the test case input file.

12. (Canceled).

13. (Currently Amended) The method of claim 8 wherein the ~~test commands~~ case input file can be is created offline and subsequently communicated to the interpretive engine.

14. (Currently Amended) The method of claim 8 wherein any of the test software test tools can be removed and replaced with another test software test tool.

15. (Currently Amended) A computer readable medium including instructions stored thereon which when executed cause the computer to perform the steps of:

executing a software application source code, wherein the software application source code defines a software application under development, including a graphical user interface as part of the software application, and wherein the software the software application source code executes to display its graphical user interface;

providing one or more software test tools that are invoked to perform testing operations on the graphical user interface that is displayed while the software application is running;

allowing a user tester to enter a test case input file stored on the computer readable medium, that contains a plurality of generic interface commands, wherein the test case input file can be edited and reused as necessary by a user to specify different generic interface commands for testing against a software application's graphical user interface in the same or a different software test tool; and number of commands via an editor or interface; and

translating, using an interpretive engine using a plurality of dynamically loaded libraries corresponding to the plurality of software test tools to receive the generic interface commands defined in the test case input file, load required libraries to map the generic interface commands to corresponding tool-specific testing operations, invoke the software test tools to perform the testing operations on the software application's graphical user interface, and report the success or failure of the testing operations , the generic interface commands received from the tester, and mapping, using a native library, the generic commands to native language understood by a particular test software tool;

wherein the interpretive engine uses the native library to map the into tool dependent codes that are then passed to the test software tool,

wherein libraries can be loaded to support different test software suites and applications.

16. (Currently Amended) The computer readable medium of claim 15 wherein the system includes the test software test tools are stored locally on [[the]] a same computer or machine as the software application under development.

17. (Currently Amended) The computer readable medium of claim 15 wherein the test software ~~test tools~~ [[is]] are stored at another computer or machine as the software application under development.

18. (Currently Amended) The method of claim 8 wherein the an editor or wizard provides a graphical interface to allow the user ~~tester to enter said test commands~~ to edit or create the test case input file.

19. (Canceled).

20. (Currently Amended) The computer readable medium of claim 15 wherein the test ~~commands~~ case input file ~~can be~~ is created offline and subsequently communicated to the interpretive engine.

21. (Currently Amended) The computer readable medium of claim 15 wherein any of the test software tools can be removed and replaced with another test software tool.